edWisor

# Advanced Predictive Analytics Using R and Python

- MUQUAYYAR AHMED
DATA SCIENTIST

# We learnt!!!

edWisor

## Statistics

## Predictive Analytics

## Advanced Predictive Analytics

- Introduction to Machine learning
- Decision Trees
- Error Metrics
- Random Forest
- Linear Regression
- Logistic Regression
- Visualizations
- KNN
- Naïve Bayes
- Cluster Analysis

# Today's Agenda

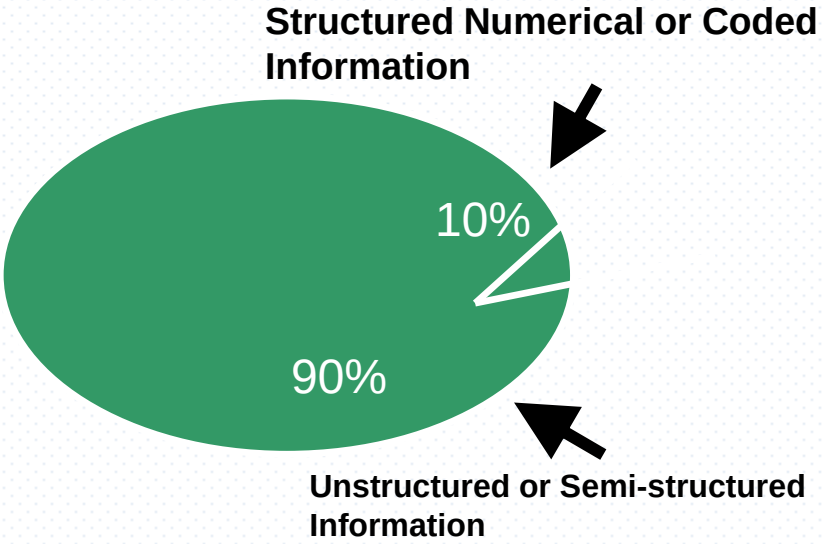
- Text Mining Analysis

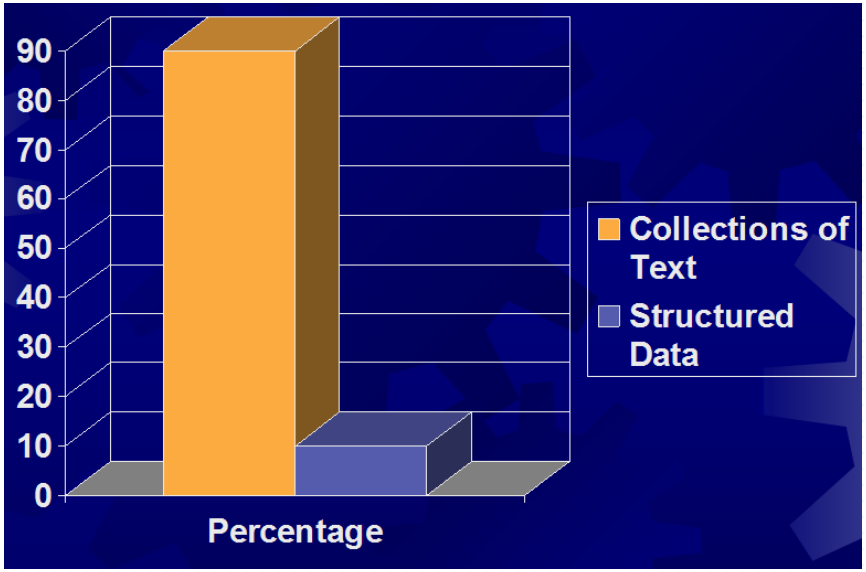
# What is Text Mining??


- Process of extracting interesting and non-trivial information and knowledge from unstructured text.
- Process of identifying novel information from a collection of texts (also known as a corpus).
- Discover useful and previously unknown “gems” of information in large text collections
- Patterns, associations, trends

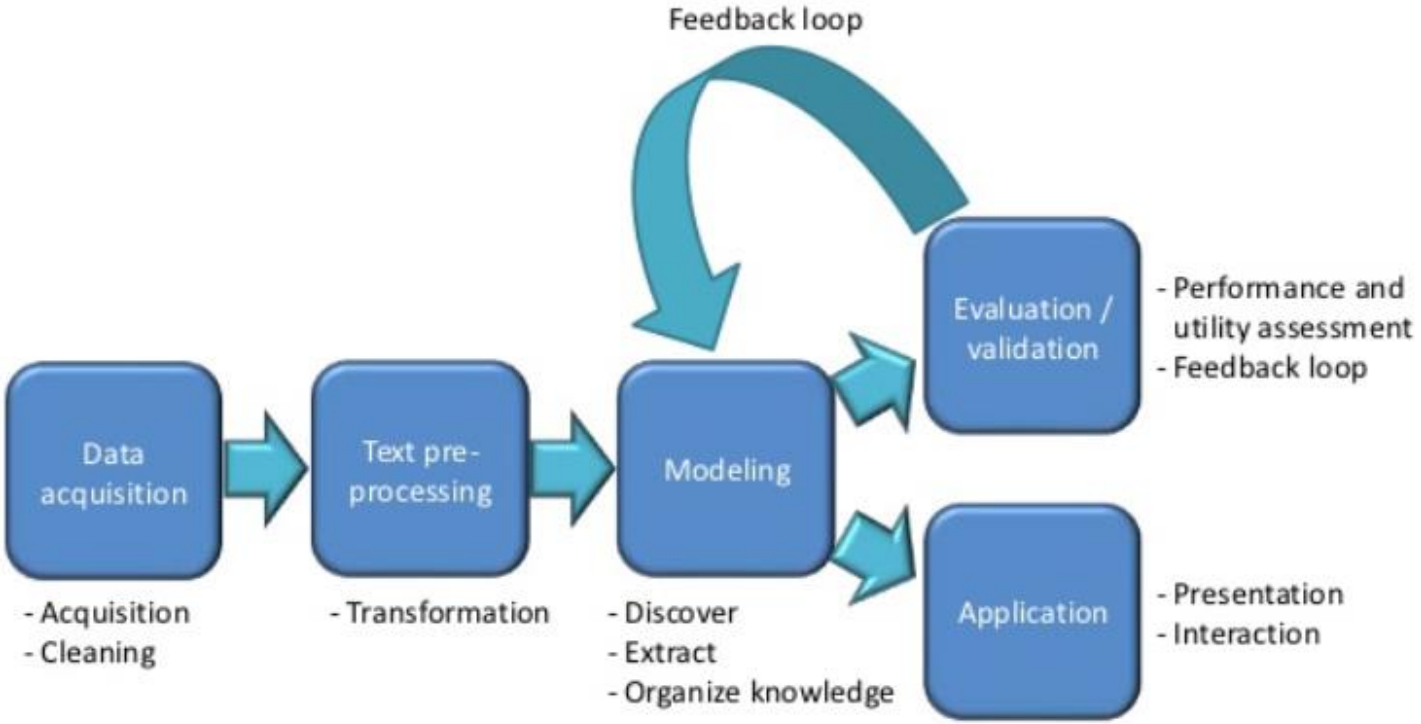
# Motivation for Text Mining

- Approximately 90% of the world's data is held in unstructured formats (source: Oracle Corporation)
- Information intensive business processes demand that we transcend from simple document retrieval to "knowledge" discovery.



# Text Mining Process



# Contd..

- Few sources of text
  - Facebook
  - Twitter
  - Yahoo
  - Google
  - Blogs
  - Watsapp etc..




# Types of problems..

Supervised learning (classification)

- The training data is labeled indicating the class
- New data is classified based on the training set
- Correct classification: The known label of test sample is identical
- with the class result from the classification model

Unsupervised learning (clustering)

- The class labels of training data are unknown
- Establish the existence of classes or clusters in the data
- Good clustering method: high intra-cluster similarity and low inter-cluster similarity

# Pre-processing steps

- Punctuation Marks
- Numbers
- Case folding
- Stop words
- White spaces
- Stemming
- Lemmatization
- Synonym check

# Pre-processing of text..

- Tokenization: Convert text into tokens
- Input: Documents that are read one by one from the collection
- Output: Tokens to be added to the index
  - No punctuation, no stop-words, stemmed

# Punctuation Marks

Remove punctuation marks

More careful approach:

- – Separate ? ! ; : “ „ [ ] ( ) < >
- – Care with .
- – Care with other punctuation marks

# Numbers

- Remove numbers
  - 3/12/91
  - Mar. 12, 1991
  - 55 B.C.
  - B-52
  - 100.2.86.144

# Case Folding

- Reduce all letters to lower case
  - exception: upper case in mid-sentence
    - e.g., General Motors
    - Fed vs. Fed
    - SAIL vs. sail

# Stopwords

- It is typical to exclude high-frequency words (e.g. function words: “a”, “the”, “in”, “to”; pronouns: “I”, “he”, “she”, “it”).
- Stopwords are language dependent

- For efficiency, store strings for stopwords in a hash table to recognize them in constant time.

- How to determine a list of stopwords?
  - For English?
    - may use existing lists of stopwords
      - E.g. SMART‟s common word list
      - WordNet stopword list
      - http://www.ranks.nl/resources/stopwords.html
    - For Spanish? Bulgarian? Hindi?

# Lemmatization

- Reduce inflectional/variant forms to base form

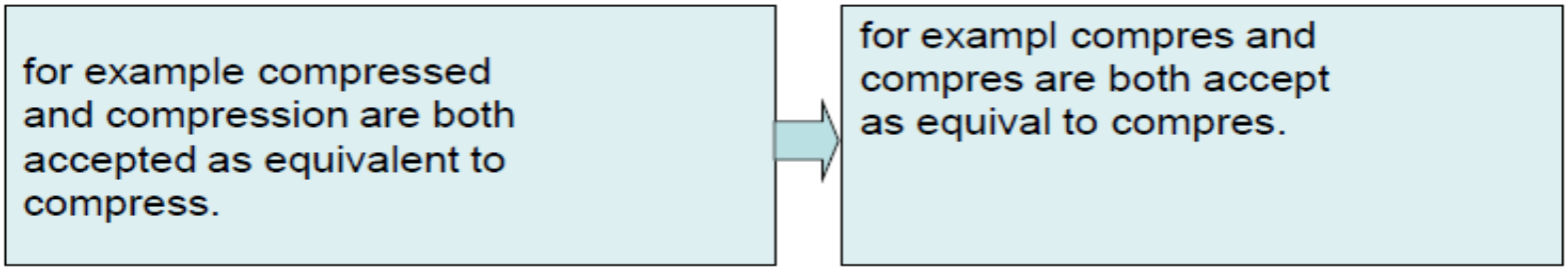
- Direct impact on VOCABULARY size
  - am, are, is  make it to ‘be’
  - car, cars, car's, cars‘ make it to ‘car’

- “the boy's cars are different colors”  make it to “the boy car be different color”

- How to do this?
  - Need a list of grammatical rules + a list of irregular words
  - Children  make to child, spoken make it to speak …
  - Practical implementation: use WordNet’s ‘morphstr’ function

# Stemming

- Reduce tokens to “root” form of words to recognize morphological variation.
  - “computer”, “computational”, “computation” all reduced to same token “compute”
- Correct morphological analysis is language specific and can be complex.
- Stemming “blindly” strips off known affixes (prefixes and suffixes) in an iterative fashion.



# HTML Tags

- Meta data
- Should text in HTML commands not typically seen by the user be included as tokens?
  - Words appearing in URLs.
  - Words appearing in “meta text” of images.

- Simplest approach is to exclude all HTML tag information (between “<“ and “>”) from tokenization. But could lose critical information.

# Term-Document Matrix

- A collection of n documents can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the “weight” of a term in the document; zero means the term has no significance in the document or it simply doesn’t exist in the document.

| | $T_1$ | $T_2$ | .... | $T_t$ |
|---|---|---|---|---|
| $D_1$ | $w_{11}$ | $w_{21}$ | ... | $w_{t1}$ |
| $D_2$ | $w_{12}$ | $w_{22}$ | ... | $w_{t2}$ |
| : | : | : | | : |
| : | : | : | | : |
| $D_n$ | $w_{1n}$ | $w_{2n}$ | ... | $w_{tn}$ |

# TF-IDF Weighting

- A typical weighting is tf-idf weighting:

  $$W = tf * idf$$

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight.

- Experimentally, tf-idf has been found to work well. It was also theoretically proved to work well (Papineni, NAACL 2001)

# Contd..

- Term Weights: Term Frequency (TF)
  - More frequent terms in a document are more important, i.e. more indicative of the topic.

  - May want to normalize term frequency(tf) across the entire corpus:

    TF = (Number of times term t appears in a document) / (Total number of terms in the document)

# Contd..

- Term Weights: Inverse Document Frequency (IDF)
  - Terms that appear in many different documents are less indicative of overall topic.
  - IDF = log10(Total number of documents / Number of documents with term t in it).

# Computing TF-IDF: An Example

- Given a document containing terms with given frequencies:

  A(3), B(2), C(1)

- Assume collection contains 10,000 documents and document frequencies of these terms are:

  A(50), B(1300), C(250)

- Then:

  A: tf = 3/6; idf = log(10000/50, 10) = 2.3; tf*idf = 1.15

  B: tf = 2/6; idf = log(10000/1300, 10) = 0.88; tf*idf = 0.29

  C: tf = 1/6; idf = log(10000/250, 10) = 1.6; tf*idf = 0.26

# Word Cloud..

Word clouds are normally used to display the frequency of appearance of words in a particular document or speech

More frequently used words appear larger in the word cloud.

The frequency is assumed to reflect the importance of the term in the context of the document.

# Contd..


BIG
DATA
LARGE
STORAGE
MILLION
SETS
SCIENCE
FUTURE
SOFTWARE
SAN
NAS
GOVERNMENT
PROCESS
VOLUME
TECHNOLOGIES
INFORMATION
MANAGEMENT
DEVELOPMENT
TECHNOLOGY
BILLION
NEW
PARALLEL
PETABYTES
ANALYSIS
RESEARCH
LEARNING
SEARCH
INTERNET
BASED
WORLD
EVERY
AMOUNT
APPLICATIONS
EXABYTES
SIZE
ANALYTICS
PROCESSING
DATABASES
FRAMEWORK
BUSINESS
SOCIAL
NETWORKS
HYPOTHESIS
INITIATIVE
ECONOMIC
TERABYTES
PRIVACY
INTERNATIONAL
IMPLEMENTATION
BECOMES
SECOND
NATIONAL
FLOW
INTELLIGENCE
APPROACHES
PROCESSES
PARADIGM
RECORDS
COST
INSIGHT
TOOLS
CENTER
FUNDING
ARCHITECTURE
SHARED
TIMES
VARIOUS
PRIVATE
TERMS
YEAR
SUCCESSFUL
NEEDED
USES
COMPARED
CAPABILITIES
LESS
HUGE
REAL
STRUCTURE
WORK
INFRASTRUCTURE
RATE
HIGHER
USERS
QUERIES
DEPARTMENT
SET
WELL
DETERMINE
ALGORITHMS
COMPANIES
VISUALIZATION
STATISTICS
SIMULATIONS
DEFINITION
BETWEEN
USING
MASSIVE
EFFECTIVE
MAKING
INTEGRATION
WORLDWIDE
DAY
ONLINE
FACTOR
MANAGE
SYSTEMS
RESULTS
YEARS
DRIVE
REPLICATION
HANDLE
SOURCES
CHALLENGES
LIMITS
COMPLEX
ANNOUNCED
CAPACITY
HAND
PAST
ONE
OUT
INCLUDE
END
MOST
NEARLY
MAPREDUCE
CRITIQUES
STATE
CRITIQUE
PEOPLE
TRAFFIC
USE
CURRENTLY
MANY
APPROACH
TIME
SENSOR
RELATED
MARKET
VARIETY
WOULD
PROJECT
ANNUAL
NOW
DISTRIBUTED
RELEVANT
USED
PER

# Text Mining Models

# Sentiment Analysis

- Sentiment analysis is the detection of attitude of a speaker with respect to some topic or overall contextual polarity of a document


- From a set of types
  - Like, love, hate, value, desire, etc.
- Simple weighted polarity:
  - positive, negative, neutral, together with strength

# Why sentiment analysis?

- Movie:  is this review positive or negative?
- Products: what do people think about the new iPhone?
- Public sentiment: how is consumer confidence? Is despair increasing?
- Politics: what do people think about this candidate or issue?
- Prediction: predict election outcomes or market trends from sentiment

# Contd..

## Text containing the attitude

- Sentence or entire document

## Simplest task:

- Is the attitude of this text positive or negative?

## More complex:

- Rank the attitude of this text from 1 to 5

## Advanced:

- Detect the target, source, or complex attitude types

# Positive or negative movie review?

- Unbelievably disappointing

- Full of zany characters and richly applied satire, and some great plot twists

- This is the greatest screwball comedy ever filmed

- It was pathetic. The worst part about it was the boxing scenes.

# Challenges in text mining

- Information is in unstructured textual form.
- Dealing with huge collections of documents
- Very high number of possible “dimensions” (but sparse)
- All possible word and phrase types in the language!!
- Complex relationships between concepts in text
    - Apple (the company) or apple (the fruit)
    - Spelling mistakes (Noisy data)